
Maschinelle Bildanalyse und Bildsynthese von Gesichtern*

Thomas Vetter

Max-Planck-Institut für biologische Kybernetik, Tübingen

Zusammenfassung

Wie lassen sich neue Bilder und Ansichten eines dreidimensionalen Objektes erzeugen, auch wenn von diesem nur ein einziges Bild aus einem anderen Blickwinkel oder unter einer anderen Beleuchtung gegeben ist. Der in dieser Arbeit vorgestellte Ansatz erwirbt Wissen um mögliche Bildveränderungen an ähnlichen Objekten und überträgt es dann auf neue Objekte. Solche auf Lernen basierte Strategien sind bei menschlichen Wahrnehmungsleistungen wohl bekannt. Im Gegensatz dazu sind im Bereich der Bildsynthese und Objekterkennung wenig konkrete Modelle vorhanden, die Aufschluß darüber geben, wie derartiges Wissen erworben werden kann, oder gar eine Implementierung auf Rechnern erlauben. In den letzten Jahren entwickelte ich das Konzept der linearen Objektklassen und implementierte eine Anwendung für menschliche Gesichter. Sie erlaubt neue Ansichten eines Gesichtes aus nur einem einzigen Photo zu erzeugen. Diese Methodik besteht einerseits aus einem allgemeinen, flexiblen Gesichtsmodell, das automatisch aus verschiedenen Beispielgesichtern erlernt wird, und andererseits aus einem Algorithmus, der es ermöglicht, dieses Modell an ein neues Gesichtsbild anzupassen. In einem ‚Analyse durch Synthese‘ Prozeß wird das neue Bild durch das Modell rekonstruiert. Mit Hilfe der zur Anpassung benötigten Modellparameter kann dann die Bildvorlage beschrieben und kodiert werden. Das Gesichtsmodell ist nun so konstruiert, daß dieser Code auch zur Synthese neuer Ansichten genutzt werden kann. Das zentrale Problem bei der Entwicklung flexibler Modelle, die alle auf der Linearkombination von Prototypen basieren, ist die Anpassung eines neuen Gesichtsbildes an das Modell: wie werden die korrespondierenden Bildpunkte zwischen den Prototypen einer

*Dieser Beitrag wurde auf Vorschlag der Jury vom Autor im Rahmen der Veranstaltung zur Verleihung des Heinz-Billing-Preises 1997 in Göttingen vorgetragen.

Objektklasse gefunden. Dieses Problem konnte bisher nur semiautomatisch gelöst werden. Hier wird ein Algorithmus vorgestellt, der vollkommen automatisch die Korrespondenz zwischen Prototypen entwickelt. Diese zum maschinellen Sehen entwickelten Objektrepräsentationen können auch für eine formale Beschreibung der menschlichen Objekterkennung genutzt werden, sowohl in Bezug auf die Organisation, als auch auf den Wahrnehmungsprozeß selbst.

1 Einleitung

Uns genügt ein Paßfoto, um eine Person wiederzuerkennen, auch wenn sie uns aus einem ganz anderen Blickwinkel begegnet. Was wir alltäglich leisten ist keineswegs selbstverständlich. Eine Richtungsänderung macht neue Gesichtspartien sichtbar, während gleichzeitig andere Bereiche plötzlich verdeckt werden. Zusätzlich ändert sich auch die Anordnung der ständig sichtbaren Gesichtspartien zueinander. Soll ein Computer eine neue Ansicht des Gesichtes aus einem einzigen Foto synthetisieren, muß er also nicht nur die unter beiden Blickwinkeln sichtbaren Bereiche an die neue Position rücken, sondern auch vorher verborgene Gesichtspartien ergänzen. Der Computer hat jedoch keinerlei direkte Information über diese Bereiche. Hätte er Kenntnisse über die allgemeine Struktur von menschlichen Gesichtern, sozusagen ein Gesichtsmodell, dann könnte er die fehlenden Gesichtspartien aus diesem Vorwissen erschließen. Ebenso könnten bildverändernde Außenfaktoren wie zum Beispiel unterschiedliche Beleuchtungen simuliert und somit kompensiert werden.

Lineare Objektklassenmodelle: Man kann einem Computer dieses Vorwissen beibringen, indem man ihm viele Aufnahmen von Gesichtern aus verschiedenen Perspektiven vorlegt und ihn so programmiert, daß er daraus die gemeinsame zugrundeliegende Struktur aller Bilder extrahiert, den Prototypen Gesicht. Generell basieren Klassenmodelle auf der Idee, ähnliche und regelmäßige Strukturen von ein und derselben Objektklasse zu erfassen und diese formell zu beschreiben. Nach diesem Prinzip ist auch das Modell der linearen Objektklassen aufgebaut (Poggio & Vetter 1992; Vetter 1996). Eine Klasse 3-dimensionaler Objekte wird als lineare Objektklasse definiert, wenn die 3-dimensionale Form der Objekte als Linearkombination einer ausreichend kleinen Anzahl von Prototypen repräsentiert werden kann. Entsprechend einer einheitlich affinen 3-dimensionalen Transformation können neue orthographische Ansichten für jedes Objekt dieser Klasse erzeugt werden. Sind die entsprechenden transformierten Ansichten der Prototypen bekannt, so kann eine rigide 3-dimensionale Transformationen genau durchgeführt werden. Besteht beispielsweise die Serie der bekannten Bilder aus Frontal- und Seitenansichten der Prototypen, dann kann die Seitenansicht eines neuen Objektes dieser Klasse aus einer einzigen frontalen Ansicht generiert werden (Abbildung 1). Prinzipiell kann jedes neue Objekt der Objektklasse als

Linearkombination der schon bekannten Objekte dargestellt werden, sobald ausreichend viele Objekte im linearen Vektorraum repräsentiert sind. Dabei kann eine neue 2-dimensionale Abbildung des Objektes sogar ohne das Wissen um seine 3-dimensionale Struktur berechnet werden (Vetter & Poggio 1997).

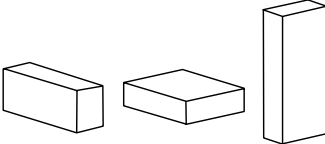
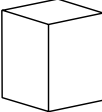
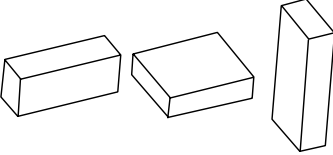
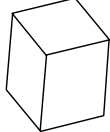
EINGABEN		
	BEISPIELE	TEST
Orientierung 1		
Orientierung 2		
		ERGEBNIS

Abb. 1: Die Rotation verschiedener 3-dimensionaler Quader einer bestimmten Orientierung (obere Reihe) in eine neue Orientierung (untere Reihe) als Beispiel für eine erlernte Bildtransformation. Der Test-Quader (obere Reihe rechts) kann als Linearkombination der 2-dimensionalen Koordinaten (2D-Form) der Beispiel-Quader aus der oberen Reihe dargestellt werden. Die Linearkombination der drei neuen Beispielansichten aus der unteren Reihe führt dann zu einer richtig transformierten Ansicht des Test-Quaders (Ergebnis untere Reihe rechts).

Bildanalyse - das Korrespondenzproblem: Lineare Objektklassen sind Modelle, die nicht nur die Formvariationen in Bildern einer gegebenen Objektklasse beschreiben, sondern prinzipiell auch die Variation von Grau- oder Farbwerten von Fotografien. Solche fotografischen Bilder werden von Computern in einzelne Bildpunkte oder Pixel aufgelöst. Um pixelbasierte Gesichtsbilder in eine lineare Objektklasse zu überführen, ist es notwendig, die Korrespondenz zwischen den Bildern zu bestimmen. Das bedeutet, daß der Computer die Pixel zwischen den vorhandenen Gesichtsaufnahmen in sinnvoller Weise miteinander verknüpfen muß. Einfaches Überlagern genügt dabei nicht: Meist liegen in verschiedenen Gesichtern Mund, Augen oder Nase nicht exakt an derselben Stelle, so daß bei einer bloßen Mittelung der Grauwerte die Gesichtsteile doppelt erscheinen (Abbildung 2). Deshalb müssen die Gesichtsbilder durch lokales Verzerren geometrisch aneinander angepasst werden. Morphing wird eines der gängigen Verfahren genannt, das kor-

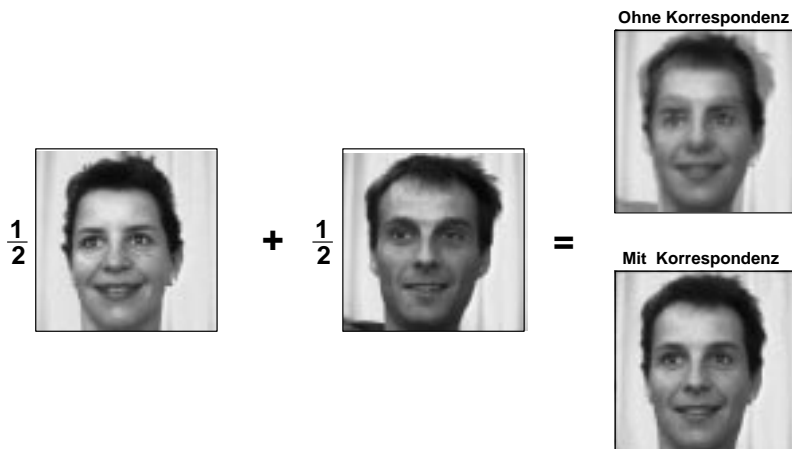


Abb. 2: Um aus zwei Gesichtsaufnahmen verschiedener Personen ein Mischgesicht zu synthetisieren, genügt es nicht, die beiden Bilder einfach zu überlagern. Korrespondierende Bildpunkte decken sich in den meisten Fällen nicht und so entstehen Doppelbilder (rechts oben). Werden korrespondierende Bildpunkte aufgefunden und einander zugeordnet, dann kann eine gewichtete Bildaddition durchgeführt werden. Dabei werden nicht nur die Grauwerte, sondern auch die Positionen entsprechender Punkte miteinander verrechnet (rechts unten). Der Computer findet die sich entsprechenden Bildpunkte automatisch.

respondierende Bildpunkte wie Nasenspitze und Mundwinkel aufsucht und dann aufeinander abbildet (Craw & Cameron, 1991; Beier & Neely, 1992).

Solche markanten Punkte zu identifizieren und einander zuzuordnen, ist maschinell nach wie vor eine äußerst schwierige Aufgabe. Bei allen bisherigen Anwendungen konnte die Korrespondenzbestimmung zwischen den verschiedenen Prototypen nur zum Teil automatisiert werden (Craw & Cameron, 1991; Lanitis et al., 1994; Hallinan, 1995; Beymer & Poggio, 1996). Entweder wurde die Korrespondenz zwischen den Bildern rein manuell bestimmt oder es wurden nur die Prototypen mit in das Modell aufgenommen, für die ein automatisches Verfahren erfolgreich war. In dieser Arbeit soll ein neues Verfahren vorgestellt werden, das mit Hilfe eines sogenannten 'Bootstrapping'-Algorithmus vollkommen automatisch die Korrespondenz zwischen den Beispielbildern bestimmen kann. Unter Aufrechterhaltung von Nachbarschaftsbeziehungen werden die Bilder solange verzerrt und deformiert, bis eine größtmögliche Ähnlichkeit zwischen ihnen erreicht ist. Dieses Verfahren ist nicht zuletzt durch den enormen Geschwindigkeitszuwachs bei der Hardware möglich.

Synthese neuer Gesichtsbilder: Setzt man auf diese Weise nicht nur zwei, sondern viele Gesichter in Korrespondenz, kann man sie auf der Basis der oben beschriebenen flexiblen Objektklassenmodelle beliebig linear kombi-

nieren und so eine Unzahl neuer Varianten erzeugen. Ist einmal die Korrespondenz zwischen den Bildern einer Objektklasse hergestellt, so lassen sich diese Bilder in zwei Vektoren aufspalten. In einen Formvektor, der die äußere Gestalt des dargestellten Objektes beschreibt, und in einen Texturvektor, der die Grau- bzw. Farbinformation trägt. Setzt man alle Beispielbilder (Prototypen) in Relation zu einem einzigen Referenzbild, das auch den Ursprung des Vektorraumes bildet, dann lassen sich diese Prototypen durch ihre Abweichung in Form und Textur zu diesem Referenzobjekt beschreiben. Durch die getrennte lineare Variation der Abweichungen zwischen den Basisbildern können viele neue Bilder aus den wenigen gegebenen Prototypen synthetisiert werden. Im Prinzip läßt sich damit sogar jedes beliebige Objekt einer Klasse darstellen, sofern die verwendete Basis der Prototypen vielfältig genug ist. In welchem Ausmaß der einzelne Prototyp zu einem neuen Objekt der Klasse beiträgt, wird durch Gewichtsfaktoren festgelegt.

In einem zweiten Teil dieser Arbeit möchte ich zeigen, wie diese Art der Bildsynthese die Möglichkeit eröffnet, aus einer einzigen Ansicht eines Gesichtes eine neue Ansicht desselben zu erzeugen. Angenommen, die bildbeschreibenden Gewichtsfaktoren sind unabhängig von der Blickrichtung, dann kann man sie für das vorliegende Gesicht ermitteln und sie anschließend bei der Synthese desselben Gesichtes in einer neuen Orientierung verwenden. Die einzige Voraussetzung dabei ist, daß, gemäß der linearen Objektklassen, die Bilder der Prototypen in beiden Orientierungen zur Verfügung stehen (vergleiche Abbildung 1). Prinzipiell benötigt man dazu nicht einmal das explizite Wissen um die 3-dimensionale Struktur des menschlichen Kopfes.

2 Automatisches Erlernen eines linearen Objektklassenmodells

Flexible Objektklassenmodelle, wie sie in der Einleitung beschrieben wurden, basieren auf der Linearkombination von Beispielbildern, die als Prototypen dienen. Mithilfe dieser Modelle kann man neue Bilder einer Objektklasse sowohl analysieren als auch synthetisieren. Soll ein spezifisches, flexibles Modell einer Objektklasse aufgebaut werden, so ist das Hauptproblem die korrespondierenden Bildpunkte zwischen den Prototypen zu bestimmen. Diese Aufgabe kann bis heute nur halbautomatisch durchgeführt werden. Halbautomatisch bedeutet, daß der Benutzer zumindest einige korrespondierende Bildpunkte in den Prototypen definieren muß, bei Gesichtern zum Beispiel Merkmale wie die Nasenspitze oder die Augenwinkel. In diesem Abschnitt beschreibe ich zum einen, wie das lineare Objektklassenmodell aufgebaut wird und zum anderen stelle ich einen Algorithmus vor, der die Korrespondenz zwischen den Prototypen vollautomatisch berechnen kann.

2.1 Erstellen eines linearen Objektklassenmodells

Um aus einem gegebenen Satz von Beispielbildern ein lineares Modell zu entwickeln, ist es notwendig, die Korrespondenz zwischen einem Referenzbild und allen Beispielbildern herzustellen. Das hier vorgestellte Verfahren zur Korrespondenzbestimmung setzt sich aus zwei Komponenten zusammen. Einem Algorithmus zur Bestimmung des optischen Flusses, der für jeden einzelnen Bildpunkt eines Gesichtsbildes die Entsprechung in einem anderen Gesichtsbild findet und somit die Korrespondenz approximiert. Diese Operation wird als Gradientenverfahren zur Bestimmung des Verschiebe- bzw.-Korrespondenzfeldes bezeichnet. Die zweite Komponente ist das lineare Objektklassenmodell selbst. Es nutzt das bereits bekannte Verschiebefeld für die Korrespondenzberechnung zu neuen Prototypen.

2.1.1 Formale Beschreibung des linearen Objektklassenmodells

Zur mathematischen Beschreibung eines linearen Objektklassenmodells sind folgende Notationen nötig. I_0, I_1, \dots, I_M sind die M Beispielbilder I_i einer Objektklasse. I_0 bezeichnet das Referenzbild der Klasse, auf das sich sämtliche Korrespondenzberechnungen beziehen. Positionen innerhalb des Referenzbildes I_0 werden mit (u, v) bezeichnet. Die pixelbasierten Korrespondenz- oder Verschiebefelder s_j zwischen I_0 und jedem der Beispielbilder I_i sind jeweils ein Abbildung $s_j : \mathcal{R}^2 \rightarrow \mathcal{R}^2$. Die Abbildung $s_j(u, v) = (x, y)$ weist jedem Punkt (u, v) in I_0 die Punkte (x, y) in den I_i zu. Diese Korrespondenzfelder s_j beschreiben die Positionen der Punkte in den Beispielbildern und werden im folgenden auch häufig als Form-Vektoren bezeichnet. Durch Abbilden der I_i auf das Referenzbild I_0 durch s_j erhält man die sogenannten Textur-Vektoren \mathbf{t}_j wie folgt:

$$\mathbf{t}_j(u, v) = I_j \circ s_j(u, v) \Leftrightarrow I_j(x, y) = \mathbf{t}_j \circ s_j^{-1}(x, y).$$

Die Menge $\{\mathbf{t}_j\}$ der Textur-Vektoren sind auch als die formnormierten Beispielbilder zu verstehen. Formnormiert im Sinne, daß die Textur-Vektoren alle die Form des Referenzbildes haben (siehe auch rechte Spalte Abbildung 5).

Das lineare Objektklassenmodell ist nun die Menge aller möglichen Modellbilder I^{modell} , die sich durch beliebige Linearkombination der Beispiel Form- und Textur-Vektoren bilden lassen. Das Modell läßt sich wie folgt über die Linearkoeffizienten $\vec{c} = [c_0, c_1, \dots, c_M]$, $\vec{b} = [b_0, b_1, \dots, b_M]$ parametrisieren

$$I^{modell} \circ \left(\sum_{i=0}^M c_i s_i \right) = \sum_{j=0}^M b_j \mathbf{t}_j. \quad (1)$$

Die Summe $\sum_{i=0}^M c_i s_i$ beschränkt die Form eines jeden Modellbildes auf Linearkombinationen der Beispiel Form-Vektoren. Gleichermaßen werden die Textur-Vektoren des Modells durch die Summe $\sum_{j=0}^M b_j t_j$ beschränkt.

Für jeden Parameterwert c_i und b_i läßt sich ein Modellbild berechnen, indem man für jeden Punkt (u, v) zuerst $(x, y) = \sum_{i=0}^M c_i s_i(u, v)$ und $g = \sum_{j=0}^M b_j t_j(u, v)$ bestimmt. Das Bild entsteht dann durch die Zuordnung $I^{modell}(x, y) = g$, eine Abbildungsvorschrift, die auch häufig als ‘Morphen’ bezeichnet wird.

2.1.2 Approximation eines Bildes durch ein lineares Objektklassenmodell

Um das Bild eines neuen Objektes I^{neu} durch ein Modell optimal zu beschreiben, müssen die Parameter des Modells so gewählt werden, daß das erzeugte Modellbild das Original möglichst genau wiedergibt. Die Genauigkeit wird über eine Fehlerfunktion definiert, die bezüglich der Parameter (c_i , b_j) zu optimieren ist (Vetter et al. 1997). Das einfachste Fehlermaß ist die L2-Norm, die wie folgt implementiert wurde.

$$E(\vec{c}, \vec{b}) = \frac{1}{2} \sum_{x,y} [I^{neu}(x, y) \Leftrightarrow I^{modell}(x, y)]^2$$

Um I^{modell} zu berechnen (siehe Gleichungen 1), muß die Formtransformation ($\sum c_i s_i$) invertiert werden, oder man arbeitet in dem Koordinatensystem (u, v) des Referenzbildes, was wesentlich effizienter ist. Deshalb wird die Formtransformation, entsprechend der aktuellen Schätzwerte für \vec{c} und \vec{b} , auf beide I^{neu} und I^{modell} angewandt. Aus Gleichung 1 folgt dann

$$E = \frac{1}{2} \sum_{u,v} [I^{neu} \circ (\sum_{i=0}^M c_i s_i(u, v)) \Leftrightarrow \sum_{j=0}^M b_j t_j(u, v)]^2.$$

Zur Minimierung der Fehlerfunktion $E(b, c)$ verwenden wir derzeit ein stochastisches Gradientenverfahren, das von Viola (1995) entwickelt wurde. Da die Anzahl der Parameter sehr groß ist (> 200) und die Optimierung über jeden Bildpunkt (256x256) berechnet wird, ist ein schnelles Verfahren zur Auswertung der Fehlerfunktion erforderlich. Das Verfahren ist nicht nur schnell, sondern auch in der Lage, lokalen Minima zu entkommen, die durch das nichtlineare Verhalten der Fehlerfunktion auftreten.

2.1.3 Gradientenverfahren zur Verschiebefeldbestimmung

Für einige Beispielbilder kann die pixelbezogene Korrespondenz zu einem Referenzbild mit Algorithmen zur Bestimmung des optischen Flußes erstellt werden (Bergen et al, 1992). Der von uns verwendete Algorithmus wurde von

Bergen und Hingorani (1990) beschrieben. Dieser Algorithmus beruht auf der Annahme, daß die Helligkeit im Bild konstant bleibt und sich nur die Lage der einzelnen Pixel zwischen Referenzbild und Beispielbild ändert. Darüber hinaus verwenden Bergen und Hingorani die Nebenbedingung, daß das resultierende Verschiebefeld glatt sein soll. Für jeden Punkt (x, y) im Bild I wird folgender Fehlerterm, $E = \sum (I_x \delta x + I_y \delta y \leftrightarrow \delta I)^2$, nach $\delta x, \delta y$ minimiert. I_x, I_y sind die räumlichen Ableitungen der Bildintensität, δI ist der Intensitätsunterschied der zu vergleichenden Bilder. Über eine Auflösungs-
pyramide werden die Verschiebungen von Auflösungsstufe zu Auflösungsstufe verbessert. Das Ergebnis dieser Berechnungen ergibt das Verschiebefeld s , welches für jeden Punkt im Referenzbild die Verschiebungen $(\Delta x, \Delta y)$ zu dem korrespondierenden Pixel im entsprechenden Beispielbild angibt.

2.2 *'Bootstrapping-Algorithmus' zur Korrespondenzbestimmung zwischen Prototypen*

Die Grundidee hinter diesem neuen Ansatz ist eine Kombination des linearen Modellansatzes mit dem Gradientenverfahren zur Verschiebefeldbestimmung (Abbildung 3): Angenommen, es sind drei Bilder einer Objektklasse gegeben, ein Referenzbild und zwei Prototypen, wobei die Korrespondenz vom Referenzbild zu einem der Prototypen, das Verschiebefeld, bereits richtig bestimmt ist. Dieses Verschiebefeld und das Referenzbild werden zu einem linearen Objektmodell vereint. Mit diesem Modell, bestehend aus Referenzbild und Verschiebefeld zu einem Prototypen, kann ein zweiter Prototyp mit Hilfe eines Optimierungsverfahrens approximiert werden. Dabei wird folgendermaßen vorgegangen: das Verschiebefeld wird solange verändert, bis ein Bild entsteht, das dem zweiten Prototypen am nächsten ist. Dadurch ist der neue Prototyp als Approximation in denselben Modellraum abgebildet, womit sich die Korrespondenzbestimmung zwischen Referenzgesicht und dem zweiten Prototypen auf die Korrespondenzbestimmung zwischen dem ursprünglichen Prototypen und seiner Approximation im Modellraum vereinfacht. Die endgültige Korrespondenz zwischen zweitem Prototypen und Referenzbild setzt sich dann aus der Kombination der zwei Verschiebefelder zusammen, aus dem zwischen Prototyp und Approximation und dem zwischen Approximation und Referenzbild. Ausgehend von einem Referenzbild und ein paar wenigen richtig bestimmten Verschiebefeldern läßt sich somit die Korrespondenz über einen ganzen Datensatz ausdehnen.

2.3 *Implementierung und Anwendung auf Gesichtsbilder*

Alle 130 Gesichtsbilder unserer Datenbank wurden wie folgt in Korrespondenz gesetzt. In einem ersten Schritt wurde ein Referenzgesicht ermittelt:

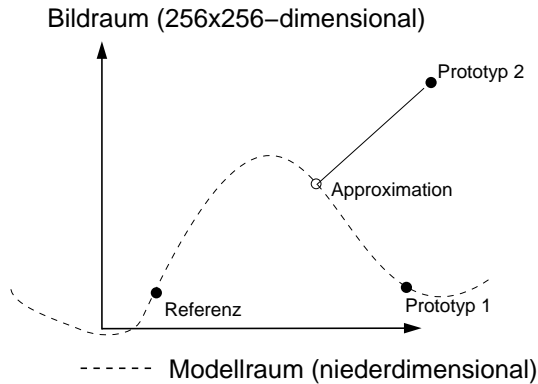


Abb. 3: Basiskonzept zur Korrespondenzbestimmung mit Hilfe eines linearen Modells. Das Verschiebefeld bzw. die Korrespondenz zwischen Referenzbild und Prototyp 1 ergibt ein lineares Modell, das im Bildraum eine eindimensionale Mannigfaltigkeit darstellt. Wird ein neues Bild derselben Objektklasse (Prototyp 2) mit dem Referenzbild in Korrespondenz gesetzt, genügt es, die Korrespondenz von Prototyp 2 zu einem Bild im Modellraum zu bestimmen. Als Bild im Modellraum wird die Approximation des zweiten Prototypen durch das Modell verwendet. Das entspricht dem Bild innerhalb der Mannigfaltigkeit, das dem zweiten Prototypen im Bildraum am nächsten liegt.

Ausgehend von einem beliebigen Gesicht wurde zu allen anderen Gesichtern die Korrespondenz mit Hilfe des Gradientenverfahrens bestimmt oder angenähert. Aus allen Verschiebefeldern wurde ein mittleres Gesicht berechnet, das Referenzgesicht. Mehrmaliges Iterieren dieses Verfahrens konvergierte in allen Fällen zum selben synthetischen Referenzgesicht (Abbildung 4). Jedoch war die Korrespondenzen zwischen dieser Referenz und den Bildern des Datensatzes nur in 80% der Fälle korrekt. In einem nächsten Schritt wurden die Korrespondenzen mit Hilfe des oben beschriebenen 'Bootstrapping'-Verfahrens erneut bestimmt. Bei diesem Verfahren wurden die statistisch signifikantesten Korrespondenzfelder mit Hilfe einer Hauptachsenanalyse über alle möglichen Korrespondenzen ausgewählt. Der erste Iterationsschritt nahm nur die ersten zwei Hauptachsen in das Modell auf und jedes Bild wurde damit approximiert. Danach wurde, wieder mit dem Gradientenverfahren, die Korrespondenz zwischen dieser Approximation und dem ursprünglichen Bild bestimmt. Die Kombination dieser neu bestimmten Korrespondenz mit der Korrespondenz zwischen Referenzbild und Approximation, die automatisch vom Modell gegeben ist, ergibt ein verbessertes Verschiebefeld zwischen dem Bild und der Referenz. Auf der Basis dieser verbesserten Korrespondenz wurde erneut eine Hauptachsenanalyse durchgeführt. In diesem zweiten Iterationsschritt wurden dann die ersten 10 Hauptachsen in das Modell aufgenom-

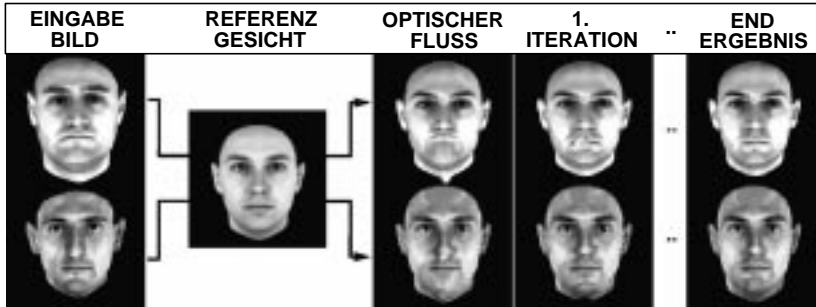


Abb. 4: Das Ergebnis verschiedener Stufen der Korrespondenzberechnung. Ist die Korrespondenz zwischen zwei Gesichtern bekannt, kann eines auf das andere abgebildet werden. Werden die Bilder der linken Spalte auf das Referenzgesicht abgebildet, entstehen die Gesichter in den rechten drei Spalten. Das Gradientenverfahren allein (optical flow) war nicht in der Lage, die zwei Gesichter der linken Spalte auf das Referenzgesicht abzubilden. Beim oberen wurde der Mund auf den Bereich zwischen Nase und Oberlippe abgebildet und beim unteren Gesicht waren Nasenspitze und Kinn falsch abgebildet. Nach zusätzlicher Iteration mit Hilfe von linearen Modellen und dem 'Bootstrapping'-Verfahren konnten diese Fehler eliminiert werden.

men. Wie oben beschrieben wurden daraufhin wiederum alle Verschiebefelder korrigiert. Nach zwei weiteren Iterationen mit 30 und 80 Hauptachsen waren die Korrespondenzen fehlerfrei und stabil (Abbildung 4).

Der vorgestellte 'Bootstrapping-Algorithmus' ist sicherlich noch nicht die vollständige Lösung des Korrespondenzproblems. Er bietet jedoch einen ersten Ansatz zur vollautomatischen Modellbildung. Der Vorteil dieses Verfahrens ist, daß es bereits erworbenes Wissen um die Korrespondenz zwischen Bildern einer bestimmten Objektklasse in die Suche nach der Korrespondenz zu einem neuen Bild miteinbringt.

3 Synthese neuer Gesichtsansichten aus einem einzigen Beispielbild

Dieser Abschnitt beschreibt die Synthese neuer Gesichtsansichten aus einem einzigen Beispielbild. Wie schon erwähnt, müssen bei der Synthese einer neuen Ansicht eines Gesichtes aus nur einem Beispielbild die Gesichtspartien, die unter beiden Blickwinkeln sichtbar sind, rekonstruiert und die vorher nicht sichtbaren Bereiche neu generiert werden. Ein 3-dimensionales Kopfmodell unterstützt die Rekonstruktion und der Ansatz der linearen Objektklassen die Ergänzung.

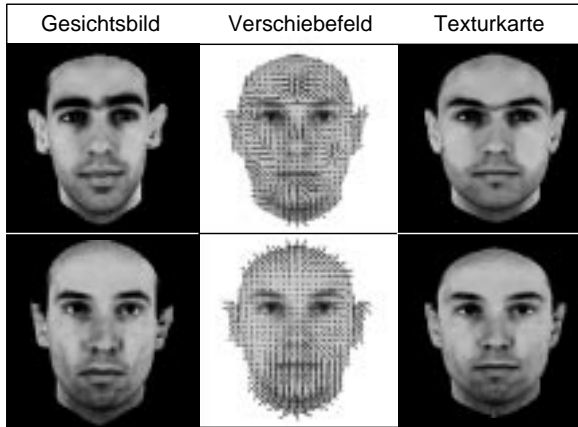


Abb. 5: Mit Hilfe einer Korrespondenzbestimmung auf der Basis einzelner Bildpunkte werden zwei Beispielgesichter (linke Spalte) auf das Referenzgesicht abgebildet (rechte Spalte). Die mittlere Spalte zeigt die Übereinanderlagerung des Referenzgesichtes mit dem jeweiligen Verschiebefeld, das durch das Gradientenverfahren berechnet wird. Die Korrespondenzberechnung trennt die 2-dimensionalen Forminformation des Bildes, die in der mittleren Spalte durch das Verschiebefeld dargestellt wird, von der Texturinformation, die in der rechten Spalte als Texturkarte auf dem Referenzgesicht abgebildet ist.

3.1 Überblick über die Vorgehensweise

Die Schlüsselidee für die Realisierung eines solchen linearen Objektmodells für Gesichter besteht in der Unterscheidung zwischen Form und Textur: jede Gesichtsansicht wird im ‘Gesichterraum’ durch einen Formvektor und einen Texturvektor repräsentiert (vergl. Cootes et al., 1995; Beymer & Poggio, 1996). Bevor jedoch die 2-dimensionale Form von der Texturinformation getrennt werden kann, wird die Korrespondenz mit Hilfe unseres Korrespondenzalgorithmus zu einem Referenzgesicht aufgebaut. Dabei hat das Modell der linearen Objektklassen den Vorteil, daß die Korrespondenz nur innerhalb eines gegebenen Blickwinkels bestimmt werden muß und nicht über große Veränderungen des Blickwinkels hinweg. Die Seitenansicht eines individuellen Gesichtes wird nur in Korrespondenz mit dem Referenzgesicht derselben Seitenansicht gesetzt. Ist das Korrespondenzproblem einmal gelöst, dann kann man diesen Datensatz in einen Form- und einen Texturvektor trennen (Abbildung 5). Der Formvektor kodiert die 2-dimensionale Form eines Gesichtes als Deformations- bzw. Korrespondenzfeld zum Referenzgesicht, das gleichzeitig den Ursprung des linearen Vektorraumes bildet. Den Texturvektor eines Gesichtes kann man sich als Textur-Karte vorstellen, da er die Bildintensitäten auf korrespondierende Gesichtspartien des Referenzgesichtes abbildet.

Schwierigkeiten bringen individuelle Gesichtsmarkmalc wie angeborene Leberflecken, die in keinem der Beispielgesichter repräsentiert sind und somit auch nicht in Korrespondenz gesetzt werden können. Eine Synthese von Gesichtspartien, die auf dem Modell linearer Objektklassen basiert, funktioniert nur bei Merkmalen der Objektklasse, die durch die Beispielbilder gegeben sind, wie Mund oder Augen. Im Gegensatz dazu können individuelle Merkmale wie angeborene Leberflecken, in einer neuen Gesichtsansicht nicht repräsentiert werden. Aus diesem Grund wird dem Verfahren ein einzelnes 3-dimensionales Modell eines menschlichen Kopfes zugefügt. Wird die Gesichtstextur zusätzlich auf das 3-dimensionale Kopfmodell abgebildet, so kann sie aus allen Blickwinkeln 2-dimensional abgebildet werden, auch wenn einzelne individuelle Merkmale im Referenzgesicht nicht vorhanden sind (Abbildung 6; mittlere Spalte)

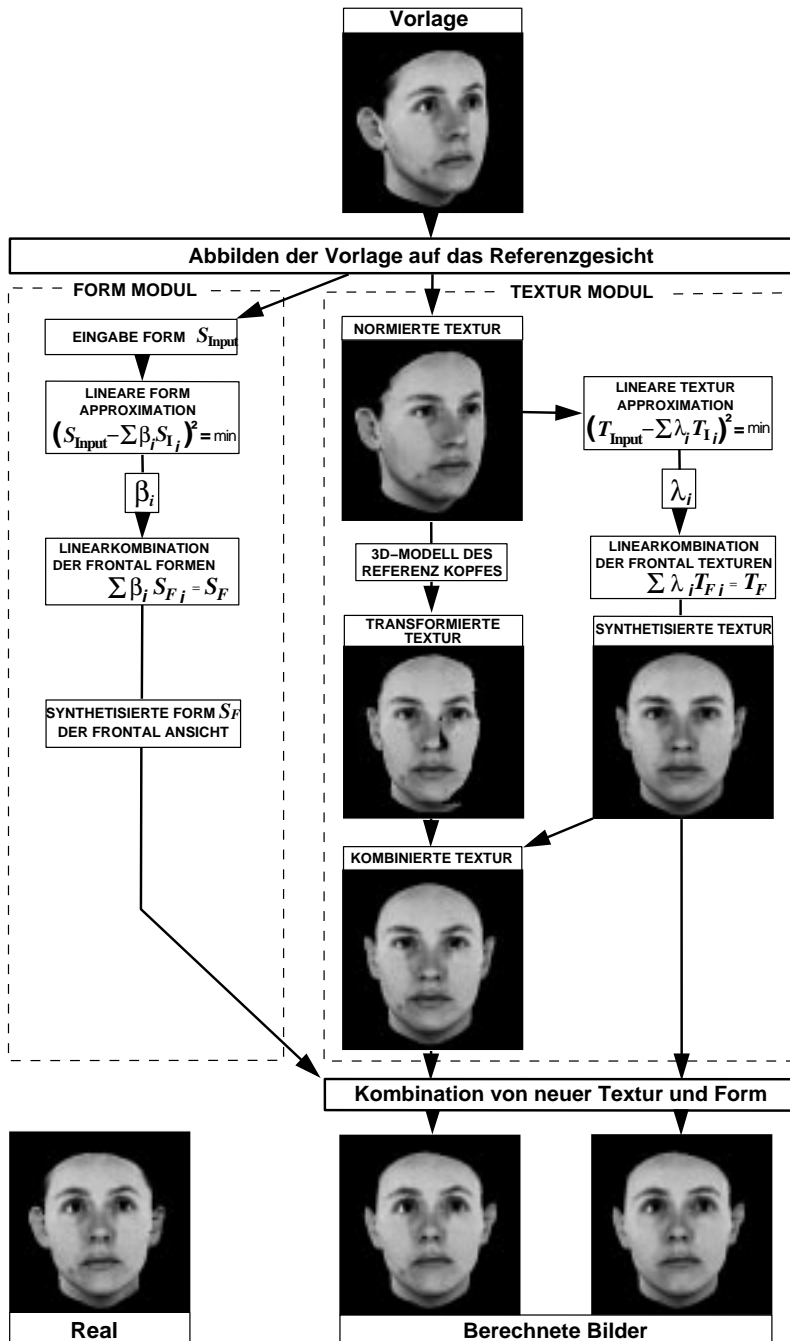
Die endgültig rotierte Version der ursprünglichen Gesichtsansicht ist dann eine Kombination aus der neuen Ansicht des 3-D Modells, individuelle Geburtsmerkmale miteingeschlossen, und der Formtransformation, die mit Hilfe der Methode linearer Objektklassen durchgeführt wurde.

3.2 *Algorithmus*

In diesem Abschnitt wird ein Algorithmus vorgestellt, der es ermöglicht, aus einem einzigen Gesichtsbild ein neues Gesichtsbild aus einem anderen Blickwinkel zu synthetisieren. Dabei wird folgendermaßen vorgegangen (Abbildung 6):

- Als erstes werden Textur- und Forminformation des Gesichtsbildes getrennt.
- Zwei separate Module bilden dann die Textur und die Form des eingegebenen Gesichtes auf das Referenzgesicht ab. Nachdem die Texturkarte und das Korrespondenzfeld erstellt sind, berechnen dieselben Module die Textur und die Form der rotierten neuen Gesichtsansicht. Beide Module

Abb. 6: Überblick über den Algorithmus, der aus einem einzigen 2-dimensionalen Eingabebild eine neue Ansicht des Gesichtes synthetisieren kann. Nachdem das Eingabebild auf das Referenzgesicht derselben Orientierung abgebildet ist, können Textur- und Forminformation getrennt weiterverarbeitet werden. Das flexible Gesichtsmodell, das aus vielen Beispielbildern erlernt wurde, ermöglicht die Vorhersage der 2-dimensionalen Form und Textur der neuen Gesichtsansicht. Die neue, noch auf dem Referenzkopf abgebildete Textur, wird entlang des Verschiebefeldes verzerrt. Das entspricht der Synthese der neuen Ansicht (untere Reihe rechts). Wird die Textur zusätzlich mit Hilfe eines allgemeinen 3-dimensionalen Modells des menschlichen Kopfes berechnet, dann kann auch die Position eines individuellen Gesichtsmarkmales abgeschätzt werden (untere Reihe, mittleres Bild). Zum Vergleich ist die echte Frontalansicht des Gesichtes unten links abgebildet.



- werden im folgenden genauer beschrieben.
- Zum Schluß wird die Textur wieder mit der Form kombiniert, das heißt die neu berechnete Texturkarte wird mittels des neu berechneten Korrespondenzfeldes zur rotierten Ansicht des Gesichtes transformiert.

Das Modul für die 2-dimensionale Formverarbeitung: Das 2-dimensionale Form-Modell des menschlichen Gesichtes basiert auf der Idee der linearen Objektklassen, wie sie in den vorangegangenen Abschnitten beschrieben wurden (notwendige und ausreichende Bedingungen des siehe (Vetter & Poggio, 1996)) und wird über einen Datensatz von Paaren verschiedener Gesichtsbilder gebildet (Abbildung 6, linke Spalte). Von jedem Bildpaar, bestehend aus einer “rotierten” und einer “frontalen” Ansicht des Gesichtes, wird der 2-dimensionale Formvektor s^r für die “rotierte” Form und s^f für die “frontale” Form berechnet. Nimmt man an, daß die 3-dimensionale Form des menschlichen Kopfes punktweise definiert werden kann, dann kann sie durch den Vektor $\mathbf{S} = (x_1, y_1, z_1, x_2, \dots, y_n, z_n)^T$ dargestellt werden, der die x, y, z -Koordinaten aller n Merkmalspunkte beinhaltet. Angenommen $\mathbf{S} \in \mathbb{R}^{3n}$ sei die Linearkombination von q 3-dimensionalen Formen \mathbf{S}_i anderer Beispielsköpfe, so daß: $\mathbf{S} = \sum_{i=1}^q \beta_i \mathbf{S}_i$. Dann folgt für jede Lineartransformation R (d.h. 3-dimensionale Rotation) mit $\mathbf{S}^r = R\mathbf{S}$, daß $\mathbf{S}^r = \sum_{i=1}^q \beta_i \mathbf{S}_i^r$. Kann die 3-dimensionale Form als gewichtete Summe der Form aller anderen Köpfe dargestellt werden, dann ist die rotierte Form die Linearkombination der rotierten Köpfe mit derselben Gewichtung β_i .

Um dies auf die 2-dimensionalen Formvektoren der Gesichtsbilder anzuwenden, muß folgendes berücksichtigt werden. P sei eine Projektion von $3D$ nach $2D$ mit $s^r = PS^r$. Unter der Annahme, daß die notwendige Anzahl q 2-dimensionaler Form-Vektoren, um $\mathbf{S}^r = \sum_{i=1}^q \beta_i \mathbf{S}_i^r$ und $s^r = \sum_{i=1}^q \beta_i s_i^r$ zu repräsentieren, sich nicht ändert, nur dann ist eine richtige Abschätzung des Koeffizienten β_i aus den Beispielbildern möglich. Die Dimension einer 3-dimensionalen linearen 2D-Form-Klasse darf sich unter der Projektion P nicht ändern. Wird zusätzlich zu dieser Projektionsbedingung angenommen, daß s^r , die 2-dimensionale Form einer gegebenen “rotierten” Ansicht, als “rotierte” Form des Beispieldatensatzes s_i^r als

$$s^r = \sum_{i=1}^q \beta_i s_i^r, \quad (2)$$

repräsentiert werden kann, dann kann die “frontale” 2D-Form s^f einer gegebenen “rotierten” Form s^r ohne \mathbf{S} berechnet werden, indem sowohl β_i der Gleichung (2) benutzt wird als auch die anderen s_i^f , die durch die Bilder des

Beispieldatensatzes mit Hilfe der folgenden Gleichung gegeben sind:

$$\mathbf{s}^f = \sum_{i=1}^q \beta_i \mathbf{s}_i^f. \quad (3)$$

Demzufolge kann eine neue 2-dimensionale Form eines Objektes auch ohne das explizite Wissen um seine 3-dimensionale Gestalt berechnet werden. Dabei muß die Korrespondenz zwischen der Gleichung (2) und der Gleichung (3) nicht bekannt sein, da die Reihen in einem linearen Gleichungssystem frei austauschbar sind.

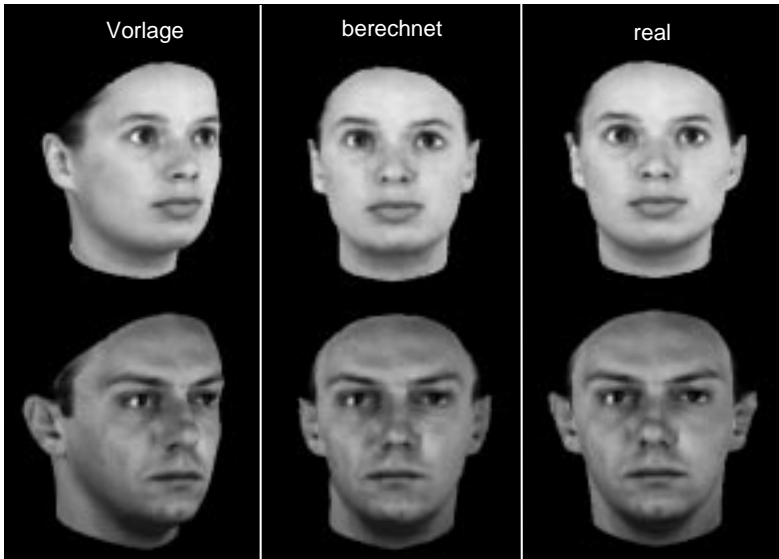


Abb. 7: Synthetisierte Frontalansichten (mittlere Spalte) eines gegebenen rotierten Gesichtes (linke Spalte). Das Vorwissen über Gesichter basierte auf einem Trainingsdatensatz von 99 Gesichtsbilder beider Orientierungen. Zum Vergleich ist die echte Frontalansicht der Gesichter in der rechten Spalte abgebildet.

Das Modul für die Texturberechnung: Im Gegensatz zur Formverarbeitung kann die neue Textur auf zwei verschiedenen Wegen berechnet werden (Abbildung 6, mittlere und rechte Spalte). Die Methode, die in der rechten Spalte von Abbildung 6 dargestellt ist, basiert auf dem Modell der linearen Objektklassen. Die Methode, die in der mittleren Spalte dargestellt ist, verwendet ein einziges 3-dimensionales Kopfmodell, um die Textur der “rotierten”

Ansicht auf die “frontale” Ansicht abzubilden (genaue Beschreibung des 3D Kopf-Modells in Vetter, 1996). Die erste Methode (rechte Spalte) entspricht der Berechnung des 2-dimensionalen Form-Vektors. Eine “rotierte” Textur \mathbf{t}^r kann als die q “rotierten” Texturen \mathbf{t}_i^r dargestellt werden, die aus dem gegebenen Beispieldatensatz folgendermaßen berechnet werden: $\mathbf{t}^r = \sum_{i=1}^q \alpha_i \mathbf{t}_i^r$. Die neue Textur \mathbf{t}^f wird dann aus den “frontalen” Beispiltexturen gebildet, indem die schon berechneten Gewichte α_i wie folgt verwendet werden: $\mathbf{t}^f = \sum_{i=1}^q \alpha_i \mathbf{t}_i^f$.

3.3 Ergebnisse

Der Algorithmus wurde an 100 Gesichtern getestet. Für jedes Gesicht standen Bilder aus zwei Orientierungen (30° und 0°) zur Verfügung. Die Korrespondenz wurde für beide Orientierungen getrennt und vollautomatisch berechnet. Die Fehler der berechneten Korrespondenz waren minimal, so daß prinzipiell alle 100 Beispielbilder herangezogen werden konnten, um ein flexible Gesichtsmodell aufzubauen. Dann wurde eine Seitenansicht (30°) eines Gesichtes ausgewählt, zu der eine Frontalansicht (0°) synthetisiert wurde. Die jeweils verbleibenden 99 Gesichtspaare (30° und 0° Ansicht) wurden verwendet, um das 2D-Form- und Textur-Modell des menschlichen Gesichtes zu erlernen. Abbildung 7 zeigt das Ergebnis für 2 Gesichter. Als Qualitätstest für die Bildsynthese wurden 10 Versuchspersonen gebeten, zwischen der echten und der synthetischen Frontalansicht zu unterscheiden. Nur 6 der 100 synthetischen Frontalansichten konnten als solche identifiziert werden. Bei allen anderen war mindestens eine Versuchsperson nicht in der Lage, zwischen realem und virtuellem Bild zu unterscheiden.

4 Aussichten und Anwendungsbereiche

Erlernt der Computer ein flexibles Gesichtermodell, so kann er sein Wissen sowohl zur Analyse unbekannter Gesichtsaufnahmen als auch zur Synthese virtueller neuer Gesichter nutzen. Dabei wird jedes individuelle Gesicht in einem linearen Gesichterraum als Vektor dargestellt. Als Ursprung des Raumes kann man sich ein aus allen Gesichtern gemitteltes Referenzgesicht vorstellen, ein Durchschnittsgesicht. Jedes einzelne Gesicht unterscheidet sich dann nur in einem individuellen Vektor von diesem Durchschnitt. Ebenso kann man ein durchschnittliches Männergesicht, als Mittelwert aller männlichen Gesichter, und ein mittleres Frauengesicht erzeugen, durch die eine ‘Geschlechtsachse’ im Gesichterraum verläuft. Der Faktor, in dem sich diese beiden Mittel unterscheiden, ist dann der Unterschied zwischen dem, was ein Gesicht weiblich macht und dem, was es männlich macht (O’Toole et al.,

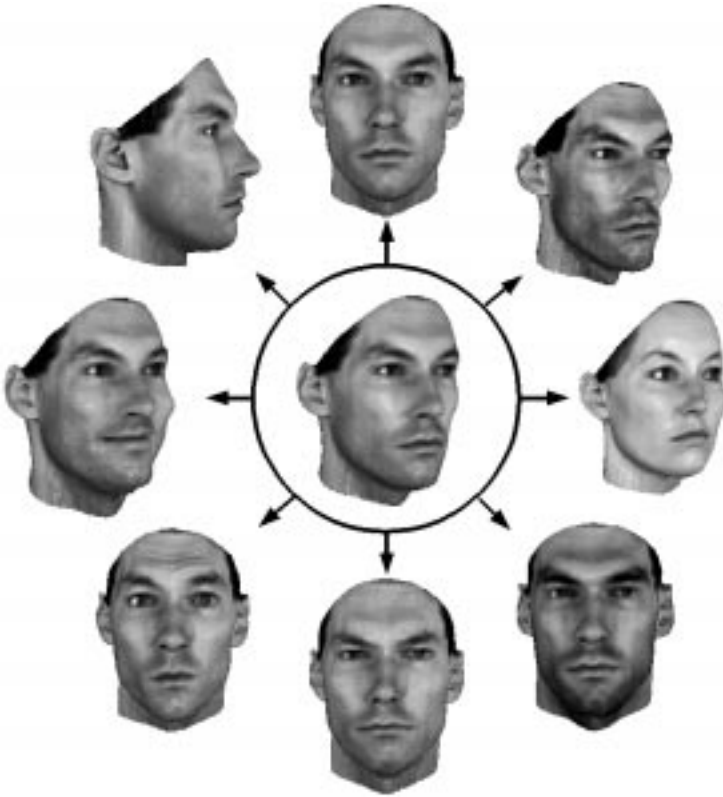


Abb. 8: Mögliche Anwendungen flexibler Gesichtsmodelle. Neben der Synthese neuer Ansichten aus einem einzigen Beispielgesicht (oben links), kann auch das Geschlecht (rechts) sowie die Mimik (unten links) künstlich verändert werden.

1997a). Entlang dieser Achse kann durch Verschiebung der 'Geschlechtsge-
 wichte' jedes individuelle Gesicht männlicher oder auch weiblicher gemacht
 werden (Abbildung 8). Oder das individuelle Gesicht wird im Gesichterraum
 einfach vom mittleren Referenzgesicht entfernt, seine individuellen Merkmale
 sozusagen verstärkt (O'Toole et al., 1997b). In Abbildung 8 ist dargestellt,
 welche neuen Möglichkeiten dieser Ansatz liefert. Sogar die Mimik von Ge-
 sichtern ist mit Hilfe eines flexiblen Gesichtermodelles auf jedes individuelle
 Gesicht übertragbar. So kann, wie der Maler mit seinem Grundfarben, mit
 Hilfe dieses Gesichterraumes jedes neue Gesicht gemischt werden, vorausge-
 setzt, die Basis der Beispielgesichter ist groß genug. Sollen Gesichter anderer
 Altersgruppen oder Rassen generiert werden, müssen selbstverständlich auch
 Beispiele dieser Personengruppen vorhanden sein.

Ist einmal ein flexibles Modell einer Objektklasse aufgebaut, so kann es für die verschiedensten Bereiche eingesetzt werden. Diese Methode findet ihre technische Anwendung nicht nur im Bereich der Computergrafik und Computervision, sondern ist auch zum Beispiel für das Anfertigen von Fahndungsbildern interessant. Ist von einer gesuchten Person nur die Fotografie einer älteren Seitenansicht mit fröhlichem Gesichtsausdruck vorhanden, könnte diese auch mit ernster Miene aus der Frontale dargestellt werden. Auch für die Filmindustrie ist dieser Ansatz von großer Bedeutung. Mit dieser Morphtechnik könnte man prinzipiell aus einzelnen Aufnahmen eines Filmstars komplett bewegte Filmsequenzen herstellen. Diese Anwendung bleibt momentan noch eine Vision. Das Verfahren funktioniert gut mit Gesichtsaufnahmen, die unter Laborbedingungen bei konstanter Beleuchtung und exakter Ausrichtung aufgenommen wurden. Das Verfahren ist noch nicht stabil genug, um mit beliebigen Fotografien zu arbeiten.

Trotzdem kann sie in einem zweiten Anwendungsbereich schon voll eingesetzt werden: im Bereich der menschlichen Objekterkennung. Hier liefert die Idee flexibler Objektmodelle nicht nur einen möglichen Ansatz, wie die Gesichtererkennung beim Menschen organisiert sein könnte, sondern kann auch zur Untersuchung des Wahrnehmungsprozesses selbst eingesetzt werden (O'Toole et al., 1997a,b).

Literatur

- Beier, T. and Neely, S. (1992). Feature-based image metamorphosis. In *SIGGRAPH '92 proceedings*, pages 35–42, Chicago, IL.
- Bergen, J., Anandan, P., Hanna, K., and Hingorani, R. (1992). Hierarchical model-based motion estimation. In *Proceedings of the European Conference on Computer Vision*, pages 237–252, Santa Margherita Ligure, Italy.
- Bergen, J. and Hingorani, R. (1990). Hierarchical motion-based frame rate conversion. Technical report, David Sarnoff Research Center Princeton NJ 08540.
- Beymer, D. and Poggio, T. (1996). Image representation for visual learning. *Science*, 272:1905–1909.
- Beymer, D., Shashua, A., and Poggio, T. (1993). Example-based image analysis and synthesis. A.I. Memo No. 1431, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.
- Cootes, T., Taylor, C., Cooper, D., and Graham, J. (1995). Active shape models - their training and application. *Computer Vision and Image Understanding*, 61:38–59.
- Craw, I. and Cameron, P. (1991). Parameterizing images for recognition and reconstruction. In Mowforth, P., editor, *Proc. British Machine Vision Conference*, pages 367–370. Springer.
- Hallinan, P. (1995). A deformable model for the recognition of human faces under arbitrary illumination. Doctoral thesis, Harvard University, Cambridge, Massachusetts.
- Lanitis, A., Taylor, C., and Cootes, T. (1994). An automatic face identification system using flexible appearance models. In *Proc. British Machine Vision Conference*, pages 66–75, BMVA Press.
- O'Toole, A., Vetter, T., Bühlhoff, H., and Troje, N. (1997a). Sex classification is better with 3D head structure than with image intensity information. *Perception*, 26:75–84.
- O'Toole, A., Vetter, T., Volz, H., and Salter, E. (1997b). Three-dimensional caricatures of human

- heads: distinctiveness and the perception of facial age. *Perception*, in press.
- Poggio, T. and Vetter, T. (1992). Recognition and structure from one 2D model view: observations on prototypes, object classes, and symmetries. A.I. Memo No. 1347, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.
- Vetter, T. (1996). Learning novel views to a single face image. In *Proc. of the 2nd Int. Conf. on Automatic Face and Gesture Recognition*, pages 22–27, IEEE Comp. Soc. Press, Los Alamitos, CA.
- Vetter, T., Jones, M. J., and Poggio, T. (1997). A bootstrapping algorithm for learning linear models of object classes. In *IEEE Conference on Computer Vision and Pattern Recognition – CVPR'97*, Puerto Rico, USA. IEEE Computer Society Press.
- Vetter, T. and Poggio, T. (1996). Image synthesis from a single example image. In Buxton, B. and Cippola, R., editors, *Computer Vision – ECCV'96*, Cambridge UK. Springer, Lecture Notes in Computer Science 1065.
- Vetter, T. and Poggio, T. (1997). Linear objectclasses and image synthesis from a single example image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):733–742.
- Viola, P. (1995). Alignment by maximization of mutual information. A.I. Memo No. 1548, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.